SYSTEM AND METHOD FOR MANAGING A VoIP NETWORK

BACKGROUND

[0001] The invention relates generally to the field of telecommunications. More specifically, but not by way of limitation, the invention relates to a system and method for managing a Voice over Internet Protocol (VoIP) network.

[0002] Although systems and methods are known for managing IP networks, the known systems and methods have several disadvantages with respect to VoIP traffic. For instance, in most data networks, the timing and scheduling of data packet transmissions is not critical. For VoIP applications, however, timing is of the essence to enable near real-time conversations. As a consequence, monitoring and management tools developed for data networks don't translate well to voice applications.

[0003] What is needed is a system and method for managing IP networks that provides real-time, or near-real time, performance monitoring of Internet Protocol - Private Branch eXchanges (IP-PBX's) and/or other IP network components in a way that is relevant to voice communications. And because many IP networks are not highly reliable, systems and methods are needed that can mitigate the effects of hardware and/or software failures in a VoIP network.

SUMMARY OF THE INVENTION

[0004] The invention provides a system and method for improved management of VoIP networks. In one respect, embodiments of the invention provide a dual channel interface between a control unit and an IP-PBX. In another respect, embodiments of the invention enable the integration of IP-PBX platforms from multiple vendors by utilizing a conversion module. In yet another respect, embodiments of the invention monitor the performance of IP-PBX's, and/or their power sources, and initiates action where performance is deteriorating and/or where failures have occurred. In yet another respect, embodiments of the invention enable a robust 9-1-1 emergency (E-9-1-1) capability for VoIP applications.

[0005] The features and advantages of the invention will become apparent from the following drawings and detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Embodiments of the invention are described with reference to the following drawings, wherein:

[0007] Fig. 1 is a block diagram of a communications system, according to an embodiment of the invention;

[0008] Fig. 2 is a node diagram illustrating a sequence of communications between components of a communications system, according to an embodiment of the invention;

[0009] Fig. 3 is a block diagram of a functional architecture for a communications system, according to an embodiment of the invention;

[0010] Fig. 4 is a block diagram of a functional architecture for a communications system, according to an embodiment of the invention;

[0011] Fig. 5 is a flow diagram for a Quality of Service monitoring process, according to an embodiment of the invention;

[0012] Fig. 6 is a process flow diagram for an Internet access monitoring process, according to an embodiment of the invention;

[0013] Fig. 7 is a flow diagram for a power monitoring process, according to an embodiment of the invention;

[0014] Fig. 8 is a block diagram of a functional architecture of an emergency communications system, according to an embodiment of the invention; and

[0015] Fig. 9 is a flow diagram for an emergency communication process, according to an embodiment of the invention.

DETAILED DESCRIPTION

[0016] The invention is described with reference to a VoIP control unit. Figs. 1–4 to illustrate interfaces between one or more control units and other network components, according to embodiments of the invention. Figs. 5-7 depict various operations of the control unit. Figs. 8

and 9 illustrate a functional architecture and method, respectively, for supporting a VoIP-based E- 9-1-1 capability. While sub-headings are used below for organizational convenience, the disclosure of any particular feature is not necessarily limited to any particular sub-heading. We begin by introducing the control unit in a functional architecture.

Functional Architecture

[0017] Fig. 1 is a block diagram of a communications system, according to an embodiment of the invention. As shown therein, a console 105 is coupled to a control unit 110 via a link 130. In addition, control unit 110 is coupled to an IP-PBX 115 via links 120 and 125. IP-PBX 115 includes an Operating System (OS) 130.

[0018] The control unit 110 may include a central processing unit (CPU) (not shown), such as an Intel x86 or Intel x86 compatible device. The control unit 110 may further include disk or other storage (not shown) for storing programs and/or data. In addition, control unit 110 may have Random Access Memory (RAM)(not shown) to execute Linux or other resident OS, and to execute application programs.

[0019] Links 120 and 130 may be enabled by Ethernet ports (not shown) and each of the links 120 and 130 may use, for example, TCP/IP or other suitable network protocol. On the other hand, link 125 may be enabled by a serial port on controller 110 (not shown) and may be utilizing an asynchronous communication protocol, for example.

[0020] In operation, control unit 110 may communicate with IP-PBX 115 according to instructions provided by console 105. In addition, links 120 and 125 may handle different types of communications. For example, in one embodiment, link 120 is used as a command link for executing various administrative tasks, such as backup, restore, shut down, restart, or upgrade operations. By contrast, link 125 may be used for monitoring the status of the IP-PBX 115 or other device. For example, link 125 may supply information such as performance monitoring data or other status information from IP-PBX 115 to the control unit 110. A more detailed view of the communications between the console 105, the control unit 110, and the IP-PBX 115 is presented with reference to Fig. 2.

[0021] Fig. 2 is a sequence diagram illustrating communications between components of a communications system, according to an embodiment of the invention. As shown therein, messages are illustrated between the console 105, control unit 110, and IP-PBX 115. The messages between console 125 and control unit 110 are via link 130. The messages between control unit 110 and IP-PBX 115 are via link 120, except that completion of functions may be checked via status information received in the controller 110 from the IP-PBX 115 over link 125.

[0022] For executing typical administrative tasks via link 120, the console 105 sends a connect and token message 205 to the control unit 110. In response, the control unit 110 sends an accept message 210 to the console 105. Then, the console 105 sends a schedule request message 215 to the control unit 110. Schedule request message 215 could specify, for example, that a task be performed immediately (i.e., on demand), at a specified time, or at a specified time interval from a particular time. According to the schedule of message 215, the control unit 110 sends a request message 220 to the IP-PBX 115. Upon completion of the administrative task, the IP-PBX 115 sends a complete message 225 to the control unit 110. Finally, the control unit 110 sends a close message 230 to the console 105.

[0023] Accordingly, administrative commands such as backup, restore, and shut down, for example, may be initiated by console 105, scheduled through control unit 110, and executed in the IP-PBX 115. In other embodiments, control unit 110 manages more than a single IP-PBX 115.

[0024] Fig. 3 is a block diagram of a functional architecture for a communications system, according to an embodiment of the invention. As shown therein, a control unit 110 is coupled to each of two IP-PBX's 305 and 320, and to each of two Uninterruptible Power Supplies (UPS's) 315 and 325. IP-PBX's 305 and 320, and UPS 315 are coupled to the control unit 110, at least in part, via Ethernet 330. Coupling to the IP-PBX's 305 and 320 facilitates management of the VoIP network; coupling to the UPS's 315 and 325 facilitates management of power to network components.

[0025] As also illustrated in Fig. 3, IP-PBX 305 includes a conversion agent 310. Control unit 110 includes Ethernet driver 335, filter 340, packet analysis module 345, management module 350, conversion module 355, UPS agent 360, serial driver 365, serial driver 370, session controller 375, and Ethernet driver 380.

[0026] In operation, management module 350 may execute commands and collect data under the control of console 105.

[0027] IP-PBX 320 may be linked to control unit 110 as described above with reference to IP-PBX 115. For example, IP-PBX 320 may use Ethernet 330 as a command link, and IP-PBX 320 may send status information to the controller 110 via serial driver 370. Serial driver 370 may forward status information to the management module 350 via a data link (not shown). By contrast, IP-PBX 305 is linked to the controller 110 solely via Ethernet 330. Thus, IP-PBX 305 may use the Ethernet 330 as a command link, and for providing status information to the controller 110.

[0028] In general, IP-PBX's may be supplied by different vendors, each having a proprietary communication protocol. Fig. 3 illustrates two alternatives, which may be used in combination, for managing a multi-vendor IP-PBX environment. In a first embodiment, where it is determined that IP-PBX 320 is using a proprietary communications protocol, session controller 375 will direct communications between management module 350 and IP-PBX 320 through the conversion module 355 for transformation of communication protocol. In a second embodiment, communication protocol is transformed in the IP-PBX 305 using conversion agent 310. In this instance, the session controller 375 directs communications between the IP-PBX 305 and the management module 350 without the use of conversion module 355.

[0029] Fig. 3 also illustrates two alternative embodiments for communications with a UPS. UPS 325 is coupled to the control unit 110 via serial driver 365, where performance data for UPS 325 is processed by UPS agent 360 and forwarded to the management module 350. On the other hand, UPS 315 is coupled to the control unit 110 via Ethernet 330 and Ethernet driver 380, where the session controller 375 forwards performance data for UPS 315 to the UPS agent 360 for processing enroute to management module 350.

[0030] In addition, Fig. 3 illustrates a functional architecture for performing VoIP packet analysis. In particular, data packets may be mirrored or otherwise sampled from Ethernet 330 using Ethernet driver 335 and filter 340. Filter 340 may select packets according to type, session, origination IP address, destination IP address, or other criterion for input to packet analysis module 345. Packet analysis module 345 may determine whether the monitored packet performance (for example, loss, jitter, and/or latency) meets a predetermined Quality of Service (QoS) policy, for example.

[0031] In alternative embodiments, more, or fewer, IP-PBX's may be coupled to the control unit 110. Moreover, in alternative embodiments, no UPS's may be coupled to control unit 110. Control unit 110 may also have additional functional capabilities. For instance, the controller 110 may include drivers and/or functional modules for collecting performance data related to T1, E1, or other network access node. In embodiments of the invention, multiple control units may be coupled together, as described with reference to Fig. 4.

[0032] Fig. 4 is a block diagram of a functional architecture for a communications system, according to an embodiment of the invention. As shown therein, control unit 415 is coupled to link 435 via a switch 410 and a router/firewall 405. Likewise, control unit 430 is coupled to link 435 via a switch 425 and a router/firewall 420, and optional control unit 450 is coupled to link 435 via a switch 445 and a router/firewall 440. Console 455 is coupled to switch 445. Link 435 may be a Local Area Network (LAN), a Wide Area Network (WAN), or the Internet, for example. Each of router/firewalls 405, 420 and 440 may be a router, a firewall, or the combination of a router and a firewall, according to application requirements.

[0033] In a stand-alone mode, a user at console 455 may communicate with control unit 415, control unit 430, and/or optional control unit 450, for example to issue commands or receive status information. In an aggregation mode, command and status communications related to control unit 415, control unit 430, and/or optional control unit 450 are aggregated at optional control unit 450. In this instance, a user at console 455 communicates with the optional control unit 450 to manage the entire network illustrated in Fig. 4.

Monitoring and Control Functions

[0034] Figs. 5-7 depict various applications of the control units described above.

[0035] Fig. 5 is a flow diagram for a Quality of Service monitoring process, according to an embodiment of the invention. As shown therein, the process begins by monitoring performance data in step 505 and receiving traffic requirements in step 510. Monitoring step 505 may collect, for example, historical information for predetermined portions of a network, for packets originating from one or more specified start points, and/or for packets terminating at one or more specified end points. The metrics monitored by performance monitoring step 505 may include, for instance, packet loss, jitter and/or packet latency (i.e., delay). Requirements received in step 510 may specify, for example, a quality of service (QoS) policy for VoIP traffic in a portion of the network, or for traffic originating from a specified origination point. The QoS policy specified in step 510 may provide minimum standards for any one or more of loss, jitter, and/or latency.

[0036] Next, in conditional step 515, it is determined whether the monitored performance meets the QoS policy. For example, if the monitored packet loss is 1 in 100 for a particular IP origination address, but the QoS policy for that same IP origination address requires a packet loss if not greater than 1 in 10,000, then it would be determined in conditional step 515 that performance does not meet the specified QoS policy.

[0037] Where the result of conditional step 515 is positive, the process may advance to reporting step 520, then return to monitoring step 505. Where the result of conditional step 515 is negative, the process may advance to notification step 525. Notification in step 525 may be in the form of email correspondence, text message paging, or other alert. After notification step 525, the process advances to step 530 to perform active testing (e.g., network diagnostics), before continuing to reporting step 520.

[0038] In alternative embodiments, the order of notification step 525 and active testing step 530 may be switched, and results from the active testing step 530 may be included in notification step

525. In other embodiments, notification step 525, active testing step 530 and/or reporting step 520 may be omitted.

[0039] Fig. 6 is a process flow diagram for an Internet access monitoring process, according to an embodiment of the invention. In step 605 a network access point, such as a T1, E1, T3 or other node is monitored for proper operation. In conditional step 610, it is determined whether there is an error in the operation of the access node. For example, if the output of monitoring step 605 is that no signals are detected, standard T1 alarms are detected, or loop-back conditions exist, then the result of conditional step 610 may be positive.

[0040] If the result of conditional step 610 is negative, then the process returns to monitoring step 605. If however, the result of conditional step 610 is positive, then the process may advance to one or more of reboot step 615 and notification step 620. In reboot step 615, the process attempts to automatically correct the detected error, for example by rebooting an access card. In notification step 620, the process notifies a system administrator or other user via email correspondence, pager, SNMP trap, or other communication channel.

[0041] Accordingly, application code in the control unit can operate to repair errors and/or to notify a system administrator or other user of degraded network access.

[0042] Fig. 7 is a flow diagram for a power monitoring process, according to an embodiment of the invention. Power management in a VoIP network may be advantageous, for example, to preserve back-up UPS power for the most critical network resources in the event of a power utility outage. In addition, controlled shut-downs in response to a pending power outage may avoid difficult IP-PBX start-ups caused by abnormal shut-downs.

[0043] The process in Fig. 7 begins in step 705 by receiving user settings. User settings may include, for example, a number of minutes of delay that should be applied after loss of external power before one or more IP-PBX's or other resources are to be shut down by the control unit. Next, in step 710, the system monitors the power status. The system then advances to conditional step 715 to determine whether there is a loss of external power. Where the result of conditional step 715 is positive, the process advances to step 720 where the control unit shuts

down one or more resources based on the predetermined delay parameter and/or other user settings. In this instance, the process would then return to monitoring step 710.

[0044] If, however, the result of conditional step 715 is negative, the process advances to conditional step 725 where it is determined whether a power fault condition is imminent. Where the outcome of conditional step 725 is positive, the process sends alerts in step 730 and/or performs data backup in step 735 before returning to monitoring step 710. If, however, the result of conditional step 725 is negative, the process returns directly to monitoring step 710.

[0045] Thus, as illustrated in Fig. 7, applications executed by the control unit can manage power in the event of actual loss of power and/or in the case where power loss is imminent.

Emergency 9-1-1

[0046] Emergency 9-1-1 (E-9-1-1) systems typically provide call back number (including extension), geographic location information (e.g., building address and floor), and caller identification information (e.g., name and organization) to local rescue dispatchers or operators via a Public Safety Awareness Point (PSAP). Because digital VoIP phones can be easily relocated on a digital network by users, new systems and methods are needed to timely maintain accurate extension, location, and caller identification information at the PSAP.

[0047] Fig. 8 is a block diagram of a functional architecture of an emergency communications system, according to an embodiment of the invention. As shown therein, the control unit 110 is coupled to IP-PBX's 805, 810 and 815. Control unit 110 is also coupled to public safety awareness point (PSAP) updater 850 via internet 845. Control unit 110 further includes data manager 820, databases 825, 830, and 835, and Location Information Service (LIS) 840. Databases 825, 830, and 835 are coupled to the data manager 820. The data manager 820 is coupled to the LIS 840.

[0048] Database 835 associates each port on IP-PBX's 805, 810 and 815 with geographic information. Data in database 835 may be relatively static. Database 830 associates telephone extensions with caller identification information. Data in database 830 may also be relatively

static. Database 825 includes an association between extensions, locations, and caller identification information. Data in database 825 may be relatively dynamic, as phones are relocated between ports coupled to IP-PBX's 805, 810, and 815.

[0049] As will be described below with reference to Fig. 9, the data manager 820 is configured to automatically update database 825, and the LIS 840 is configured to automatically provide data to the PSAP updater 850. PSAP updater 850 provides updated extension, location, and caller identification information to a local PSAP (not shown) so that when an incoming 9-1-1 call is received by a local emergency dispatcher, accurate call-back extension, location, and caller identification information is available from the local PSAP (not shown).

[0050] Fig. 9 is a flow diagram for an E-9-1-1 process, according to an embodiment of the invention. In step 905, each IP-PBX port is associated with geographic location information (e.g., building address and floor, latitude and longitude, or other format, according to design choice). Step 910 may be performed, for example, as IP-PBX's are installed in a facility.

[0051] Next, in step 910, telephone extension are associated with caller identification information (e.g., name and organization, and/or other information). Step 910 may be performed, for instance, as employees join an organization.

[0052] Next, in discovery process 915, the extension or other identifier is captured for each VoIP phone that is newly coupled to an IP-PBX port. Step 915 may be executed by the data manager 820, for example by polling each port of IP-PBX's 805, 810, and 815. The polling in step 915 may be at predetermined intervals or at predetermined times.

[0053] Data is then resolved in step 920. For instance, where port 805-1 was associated with 123 maple street, 2nd floor in step 905, where extension 555 was associated with Tom Smith in step 910, and where it is newly discovered in step 915 that extension 555 is coupled to port 805-1, then it is resolved in step 920 that extension 555 has a location of 123 maple street, 2nd floor, and that a caller on extension 555 is Tom Smith. Step 920 may be executed by the data manager 820, for example by searching databases 830 and 835. The result of step 920 may be stored in database 825.

[0054] Finally, in step 925, newly associated data is forwarded to the PSAP updater, for example by the LIS 840. Step 925 may be performed at predetermined times, at predetermined intervals or upon updated information generated in steps 915 and 920.

[0055] Accordingly, the use of local mapping databases and the discovery process enables the control unit 110 to provide accurate call-back extension, location, and identification information to the PSAP updater 850.

CONCLUSION

[0056] The invention described above thus overcomes the disadvantages of known systems and methods by improving the performance monitoring of IP-PBX's and other IP network components in a way that is relevant to voice communications. While this invention has been described in various explanatory embodiments, other embodiments and variations can be effected by a person of ordinary skill in the art without departing from the scope of the invention.